EXTENSIONS OF THE TRAVELING SALESMAN PROBLEM: JOB SEQUENCING WITH PATTERN CONSTRAINTS OR DUE DATES

A THESIS

Presented to

The Faculty of the Graduate Division

by

Thomas D. Cash, Jr.

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in the School of Industrial Engineering

Georgia Institute of Technology

December, 1968

EXTENSIONS OF THE TRAVELING SALESMAN

PROBLEM: JOB SEQUENCING WITH

PATTERN CONSTRAINTS OR DUE DATES

Approved:

Chairman

Date approved by Chairman: Dec 12, 1968

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. C. M. Shetty who served as my Thesis Advisor. Dr. Shetty contributed a great deal of needed insight to this research effort, and his personal efforts helped to bring it to this written conclusion. I am also grateful to my reading committee members, Dr. D. E. Fyffe and Dr. L. A. Johnson, for their helpful comments.

This thesis is dedicated to my wife Camilla and my children, Scott and Kristin. Their encouragement, combined with patience and understanding, was invaluable to me during my graduate studies.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

In this report, "branch and bound" solution algorithms are developed for two types of job sequencing problems which have the sequence dependent cost structure of the standard traveling salesman problem.

The first type of sequencing problem occurs when the job sequence is required to maintain some specified pattern, i.e., the problem is subject to "pattern constraints." The requirement that the job sequence be a traveling salesman's tour is itself a pattern constraint; therefore, it is demonstrated that the branch and bound algorithms of Eastman (10) and Little, *et al.* (21), for the solution of the standard traveling salesman problem, can be extended for problems with additional pattern constraints such as precedence relations between jobs.

The second type of sequencing problem occurs when there are due dates and corresponding late charges assigned to the jobs to be processed. This problem requires the formulation of an objective function with sequence dependent change-over costs and "late penalty costs." An algorithm with the same basic structure as the algorithm of Little, *et al.* (21) is developed for the solution of this problem.

In order to state the essential structure and demonstrate the validity of these algorithms, a presentation of the general branch and bound approach is given which follows closely that of Lawler and

Wood (19). The algorithm of Eastman and Little, *et al.* are also stated in terms of this approach.

CHAPTER I

INTRODUCTION

For the standard traveling salesman problem, one wishes to find the minimum length closed circuit which connects n points of a linear graph, i.e., the minimum length Hamiltonian circuit. The problem derives its name from the following illustration. A salesman starting from some city wishes to visit each of the other n-1 cities once and only once, and return to his starting city. Thus we say that the salesman desires to find the minimum length tour through the n cities, and that he may not take any subtours.

Although the computational difficulties encountered in attempts to solve this problem have been extreme, the problem has generated a great deal of interest because of its practical uses. Many problems in sequencing and assignment for which the solution must be cyclic (a traveling salesman's tour) can be stated as a traveling salesman problem. For example, the problem of sequencing n jobs through one machine, with sequence dependent setup costs, can be formulated as a traveling salesman problem.

However, many industrial applications for the traveling salesman problem require the consideration of additional constraints, or the introduction of modified measures of effectiveness. For the example stated above, it may also be required that certain jobs precede other jobs in the sequence; or there may be due dates assigned to the jobs to be processed with penalty charges incurred by jobs which are completed after their due dates.

These precedence constraints and due dates with related late charges are the extensions of the traveling salesman problem to be considered in this report. Specific algorithms for solving such problems are discussed later under the headings "Pattern Constraints" and "Due Dates" respectively.

Both the general branch and bound approach and the dynamic programming approach appear to be effective methods of attack for these problems. However, a literature survey, which is reviewed in Chapter II, revealed that the dynamic programming approach is restricted to general traveling salesman problems of approximately 15 cities or less because of data storage requirements. Although this research has not concerned itself with the computational details of any solution algorithm, computational feasibility for large problems has been taken into account in developing algorithms for their solutions. Therefore, the needed algorithms were developed within the framework of the general branch and bound approach which is stated in Chapter III.

Chapter IV of this report is concerned with the detailed description of the previously described extensions and the branch and bound algorithms which were developed for their solution. These algorithms are illustrated by their application to example problems in the Appendices.

CHAPTER II

LITERATURE SURVEY

A review of the scheduling and sequencing literature revealed almost no consideration of any extensions of the traveling salesman problem. Bellmore and Nemhauser (4) refer to a paper by Hatfield and Pierce (16) which they say uses ". . . branch-and-bound algorithms to solve a job sequencing problem closely related to the traveling salesman problem, but further constrained because of job deadlines to be met." The relation of their problem to the "Due Date" problem considered in Chapter IV is not known because the paper by Hatfield and Pierce could not be obtained.

Conway, *et al.* (5) provide an excellent treatment of the mathematical problems of scheduling theory, including the general traveling salesman problem. They discuss the problems of job sequencing according to due dates or precedence requirements; however, they do not consider these problems with sequence dependent setup costs. For a more recent review of sequencing theory, Elmaghraby (11) and Spinner (27) are suggested.

A number of solution methods have been proposed for the general traveling salesman problem. Some of them are not guaranteed to find optimal solutions (2,7,13,18,20,23,24,25); however, several of these approximate methods, particularly those of Lin (20), and Reiter and Sherman (23), are at the present the best compromise because of the computational difficulties encountered in trying to find exact optimal solutions for large problems.

In addition to the above methods, there are several solution procedures which can be proven to yield an optimal solution. The first such algorithm was developed by Dantzig, *et al.* (8,9). This algorithm treated the traveling salesman problem as a linear integer programming problem while the requirement that the solution be a tour was met by continually incorporating integer constraints to exclude subtours. Miller, *et al.* (22) used Gomory's method of cutting planes for the solution of several traveling salesman problems formulated as integer programming problems, and their results indicate that the convergence properties for this method are very erratic.

Dynamic programming formulations for the traveling salesman problem have been given by Bellman (3), Gonzales (15), and Held and Karp (17). The reports of Bellman, and Held and Karp mentioned that the dynamic programming approach was flexible enough to incorporate precedence constraints.

Several specialized algorithms have been developed specifically for the traveling salesman problem. One of these, by Croes (6), uses an iterative "inversion" scheme to improve upon a given starting solution. When improvements are no longer possible through inversions, the method uses a tree search routine to determine the optimal solution. Another algorithm was developed by Gilmore and Gomory (14) for the solution of a specific type of traveling salesman problem which occurs sometimes as a machine sequencing problem. However, their algorithm is restricted to those problems which use a particular distance measure that is not generally applicable.

Yet another type of solution procedure developed for the traveling salesman problem are the so-called "branch and bound" algorithms. The first algorithm of this type was developed by Eastman (10) in 1958 and later developed independently by Shapiro (26) in 1966. Eastman demonstrated that his algorithm could be extended for the traveling salesman problem with precedence constraints. Little, *et al.* (21) developed another branch and bound algorithm in 1964 which is structured somewhat differently than Eastman's. A description of the essential structure of these algorithms is given in Chapter III.

An excellent review of the status of the general traveling salesman problem can be found in Bellmore and Nemhauser (4). Their comparison of the computational success of the different algorithms indicates that the branch and bound algorithms are definitely the most powerful of the exact methods. A review of the history of the traveling salesman problem prior to 1956 is given by Flood (12).

There are two pertinent conclusions which may be drawn from a survey of this literature. First, no substantial consideration has been given to either of the extensions of the traveling salesman problem described in Chapter I. Second, the branch and bound approach to the general traveling salesman problem has a demonstrated computational superiority over the other approaches.

CHAPTER III

THE BRANCH AND BOUND APPROACH

Introduction

The two most general statements of the branch and bound approach which could be found in the literature were those of Balas (1) and Lawler and Wood (19). The statement of Balas is in a somewhat more concise mathematical form than that of Lawler and Wood. However, Balas assumes that the problem has a finite solution space. Although this may be a realistic requirement in most cases, it has not been found to be necessary in the proof of convergence which is presented in this report. Lawler and Wood develop their presentation in terms of problems to be solved, rather than in terms of the solution space, as presented by Balas, and this was found to be a more convenient presentation for the algorithms of this report.

Therefore, the statement of the general branch and bound approach given in this chapter will follow closely that of Lawler and Wood. Later in the chapter, conditions are specified which will insure finite convergence of the branch and bound algorithms developed during this study.

Statement of the Branch and Bound Approach

Suppose that one wishes to find the optimal solution X^{o*} to a problem P^o stated in the following form:

$$P^o \qquad \text{minimize } C^o(X) \tag{3.1}$$

$$\text{S.T.} \qquad X \varepsilon D^o \tag{3.2}$$

where $X = (x_1, x_2, \ldots, x_n)$, C^o is an arbitrary function, and Equation (3.2) specifies the feasible solution space for problem P^o.

The "branch and bound" approach to solving this problem is to create a set of bounding problems. Each bounding problem P^j is of the form

$$P^j \qquad \text{minimize } C^j(X) \tag{3.3}$$

$$\text{S.T.} \qquad X \varepsilon D^j \tag{3.4}$$

This "current set of problems" $P = \{j : P^j \text{ is a current bounding problem}\}$ is postulated to satisfy the following "bounding properties."

(B-1) There exists an X^{o*} such that $X^{o*} \varepsilon D^b$ for some $b \varepsilon P$. In other words, an optimal solution X^{o*} must be feasible for at least one problem P^b of the current set of bounding problems.

(B-2) $C^b(X^{o*}) \leq C^o(X^{o*})$.

If the optimality conditions specified in Theorem 1 below are satisfied by any problem P^k in the current set of bounding problems, then the solution to this problem gives the optimal solution we are seeking.

<u>Theorem 1</u>: Optimality Conditions. Let P denote the current set of bounding problems. Let problem P^k be a problem in the current set P such that

$$(0\text{-}1) \qquad C^k(X^{k*}) = \min_{j \varepsilon P} C^j(X^{j*})$$

where X^{j*} is the optimal solution to problem P^j. If P^k satisfies the following optimality conditions:

(0-2) X^{k*} is feasible for problem P^o, i.e., $X^{k*} \varepsilon D^o$

$$(0\text{-}3) \qquad C^k(X^{k*}) \geq C^o(X^{k*})^{++}$$

then X^{k*} is the optimal solution to P^o, i.e., $X^{o*} = X^{k*}$.

Proof: By definition $C^b(X^{b*}) \leq C^b(X^b)$ for all X^b, which when combined with property B-1 implies that

$$C^b(X^{b*}) \leq C^b(X^{o*}) \qquad (3.5)$$

Conditions 0-1 through 0-3 in combination with the bounding properties B-1 and B-2 yield

---

++Lawler and Wood (19) use an equality here.

$$C^b(X^{o*}) \leq C^o(X^{o*}) \leq C^o(X^{k*}) \leq C^k(X^{k*}) \leq C^b(X^{b*}) \tag{3.6}$$

The inequalities (3.5) and (3.6) can be combined to give $C^b(X^{o*}) \leq C^o(X^{o*}) \leq C^o(X^{k*}) \leq C^k(X^{k*}) \leq C^b(X^{b*}) \leq C^b(X^{o*})$. This inequality can be true only if $C^b(X^{o*}) = C^o(X^{o*}) = C^o(X^{k*}) = C^k(X^{k*}) = C^b(X^{b*}) = C^b(X^{o*})$. Hence, $X^{o*} = X^{k*} = X^{b*}$, and X^{k*} is the optimal solution to P^o. This completes the proof.

If P^k does not satisfy conditions 0-2 and 0-3, then P^k bounds problem P^o in the sense that $C^k(X^{k*}) \leq C^b(X^{o*}) \leq C^o(X^{o*})$, and the optimal solution to P^o has not yet been found. In order to find the optimal solution to P^o, we can apply the following three step branch and bound procedure beginning with step 3.

1. Create a current set of bounding problems P, go to step 2.

2. Test whether the current set of bounding problems satisfies the optimality conditions of Theorem 1 or not. If the conditions are satisfied then X^{o*} has been found; if not, go to step 3.

3. Generate a new current set of bounding problems and return to step 2.

This three step procedure describes any branch and bound algorithm. The details of forming the first set of bounding problems and constructing algorithms which will terminate in a finite number of steps shall be discussed in the next section.

Application of the Branch and Bound Approach

In practice the branch and bound approach may be used to construct an algorithm for the solution of a problem which is difficult to solve directly, such as the traveling salesman problem. The first set of bounding problems generally consists of one problem (we will call this problem P^1) which can be solved directly. The three step branch and bound procedure discussed earlier will then construct a sequence of current sets of bounding problems, which are comparatively easy to solve, until a current set is reached which satisfies the optimality conditions of Theorem 1.

The process of constructing new problems is called "branching." Branching is carried out such that if a problem P^j in the current set is replaced by a set of new problems $P^{(j)}$, the new current set of bounding problems

$$\hat{P} = (P - \{j\}) \cup P^{(j)}$$

satisfies the bounding properties B-1 and B-2. The choice of which problem within the current set to branch from will depend largely upon the capabilities of the available solution machinery. Lawler and Wood (19) provide a very good discussion of the "Strategies of Branching"; hence, they will not be considered in this paper. The strategy of branching from problem P^k, which is defined by condition 0-1, will be maintained for all of the algorithms presented in this report. That is, we will branch from that problem in the current set which has the "lowest bound."

Any particular branch and bound algorithm must be constructed with a "branching rule" for the creation of new problems which will insure finite convergence of the algorithm. A branching rule which satisfies the conditions given in the following theorem is sufficient to insure the finite convergence of the branch and bound algorithms discussed in this study.

Theorem 2: Convergence. The following conditions together with the bounding properties B-1 and B-2 are sufficient to insure that a branch and bound algorithm will converge to an optimal solution in a finite number of steps.

(C-1) Let $P = P^{(o)}$ be the first current set of bounding problems, then the union of the feasible domains for the problems P^j, $j \varepsilon P^{(o)}$ must be finite, i.e., $\bigcup_{j \varepsilon P^{(o)}} D^j$ is finite. Also the cardinality of the set $P^{(o)}$ must be finite.

(C-2) Let the set of bounding problems $P^{(k)}$ be created from problem P^k, $k \neq o$; then $D^j \subseteq D^k$ for all $j \varepsilon P^{(k)}$, and the cardinality of the set $P^{(k)}$ must be finite.

$$\text{(C-3)} \qquad X^{k*} \notin D^j, \quad \text{for all } j \varepsilon P^{(k)}$$

Proof: Conditions C-2 and C-3 establish the inductive step that in progressing from any current set P to a new current set $\hat{P}$, the feasible domain of optimization will decrease by at least one point. Condition C-1 specifies that the feasible domain and the cardinality of the first current set of bounding problems is finite. Therefore, the branching process will eliminate feasible solutions until the optimal solution X^{o*} is found in a finite number of steps. This completes the proof.

It is worth noting that these convergence conditions make no special assumptions about original problem P^o. They require only that the solution domain of the first set of bounding problems $P^{(o)}$ be finite, and that this domain not be enlarged as new problems are created. Condition C-3 insures that the solution to any problem P^k will not be feasible for any of the new problems $P^{(k)}$; thereby reducing the solution space as the branching process continues. If any problem P^k has no feasible solution, then discard that problem from the current set P since there would be no feasible solution for any problem created from P^k.

At this point, it is useful to illustrate the application of the branch and bound approach by examining the structure of the algorithms developed by Eastman (10) and Little, *et al.* (21), for solving the traveling salesman problem. In order to do this, we will first present the mathematical model for the traveling salesman problem.

The traveling salesman problem as stated in Chapter I may be formulated mathematically in the following manner. Let c_{ij} be the cost of going from any city i to city j, and let $x_{ij} = 1$ imply that city i immediately precedes city j in the tour. Thus the mathematical statement of the traveling salesman problem is as follows:

$$P^o \qquad \text{minimize } C^o(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} \qquad (3.7)$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,\ldots,n \qquad (3.8)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,\ldots,n \tag{3.9}$$

$$x_{ij} = 0 \text{ or } 1, \quad \text{all } i \text{ and } j \tag{3.10}$$

$$\text{no subtours} \tag{3.11}$$

where the constraint (3.11) is explained in Chapter I.

<u>The Algorithm of Eastman</u>

Consider the current set of bounding problems P. Let problem P^k be as defined by condition 0-1 of Theorem 1, and let this be the problem from which we will branch. For Eastman's algorithm this is of the form

$$P^k \qquad \text{minimize } C^k(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} = 1, \; j = 1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1,\ldots,n$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j$$

$$x_{ij} = 0 \text{ for } x_{ij} \in R^k$$

where R^k is some specified set of variables whose values are restricted to be zero, i.e., $x_{ij} \in R^k$ implies that $x_{ij} = 0$.

Initially the current set P consists of one problem P^1 with $R^1 = \phi$, i.e., we relax the restriction that the solution have no subtours so that P^1 is an assignment problem. This problem clearly satisfies the bounding properties B-1 and B-2 and the convergence condition C-1.

Returning to the general stage bounding problem, in Eastman's algorithm each such problem satisfies the bounding properties B-1 and B-2 and the optimality condition 0-3. The violation of condition 0-2 by problem P^k (P^k is defined by condition 0-1) is used to develop a branching rule which will give a new set of bounding problems satisfying B-1, B-2, 0-3, C-2, and C-3. The branching rule for Eastman's algorithm is given below.

If P^k violates condition 0-2, X^{k*} contains two or more subtours. Suppose that the shortest of these subtours is $S_b = \{(1,2),(2,3),(3,1)\}$ so that $x_{12} = x_{23} = x_{31} = 1$. Construct L new bounding problems, where L is the length of the subtour S_b (in this case L = 3), such that each new problem P^j, $j \varepsilon P^{(k)}$, is identical to P^k except that one and only one x_{ij} for $(i,j) \varepsilon S_b$ is required to be zero for each problem P^j. It is also required that $\bigcup_{j \varepsilon P^{(k)}} R^j = R^k \bigcup \{x_{ij} : (i,j) \varepsilon S_b\}$. Thus for $S_b = \{(1,2),(2,3),(3,1)\}$, we have three new problems with $R^j = R^k \bigcup \{x_{12}\}$ for one problem, $R^j = R^k \bigcup \{x_{23}\}$ for the second, and $R^j = R^k \bigcup \{x_{31}\}$ for the last problem.

<u>The Algorithm of Little, *et al.*</u>

Again we want to solve the traveling salesman problem P^o as defined by Equations (3.7) through (3.11). Let P^k, $k \varepsilon P$ be defined by condition 0-1 of Theorem 1. In Little's algorithm P^k is of the form

$$P^k \qquad \text{minimize } C^k(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} \leq 1, \; j = 1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} \leq 1, \; i = 1,\ldots,n$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \geq m^k$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j$$

no subtours

$$x_{ij} = 0 \text{ for } x_{ij} \in R^k$$

$$x_{ij} = 1 \text{ for } x_{ij} \in Q^k$$

where R^k is the set of variables specified to be equal to zero; Q^k is the set of variables specified to be equal to one, and the cardinality of Q^k is m^k-1. The solution to this problem is clearly to have $x_{ij} = 1$ for all $x_{ij} \in Q^k$, $x_{ij} = 0$ for all $x_{ij} \in R^k$, and $x_{pq} = 1$ for some $X_{pq} \notin Q^k \cup R^k$ while the remaining $x_{ij} = 0$.

To initialize the algorithm, the first current set of bounding problems P consists of one problem P^1 with $R^1 = Q^1 = \phi$; hence $m^1 = 1$. If $c_{pq} = \min c_{ij}$, the solution to P^1 is obviously $x_{pq} = 1$ and $x_{ij} = 0$ for all the remaining variables. Problem P^1 clearly satisfies condition C-1.

Returning to the general stage problem P^k as defined by condition C-1; this problem clearly satisfies B-1, B-2, and 0-3. The branching rule discussed below uses the violation of condition 0-2 to create two new bounding problems P^j satisfying B-1, B-2, 0-3, and the convergence conditions C-2 and C-3. The two problems P^j, $j \varepsilon P^{(k)}$, are of the same form as problem P^k stated above. For one problem we let

$$m^j = m^k, \qquad R^j = R^k \cup \{x_{pq}\}, \qquad Q^j = Q^k$$

while for the second problem we let

$$m^j = m^k + 1, \qquad R^j = R^k, \qquad Q^j = Q^k \cup \{x_{pq}\}$$

where x_{pq} is as defined above.

For the above it is clear that x_{pq} will have a specified value of zero in the first problem and one in the second problem. The solution of these problems will have some $x_{ij} = 1$, where $x_{ij} \notin R^j \cup Q^j$ for each of the two problems. The algorithm continues constructing problems in this fashion until the optimal solution to P^o is found.

Little begins his algorithm with the cost matrix for problem P^1 reduced so that there is at least one zero in every row and column of the P^1 cost matrix. As the algorithm progresses and variables are added to the sets R^j and Q^j for each problem, the cost coefficients for these variables are removed from consideration and the cost matrix is updated to maintain the reduced state. A justification of this procedure may be found in Little, *et al.* (21) or Bellmore and Nemhauser (4). Little's algorithm also incorporates a heuristic method for breaking ties when there are multiple solutions for any particular stage problem P^k. This heuristic procedure is believed to improve the convergence speed for the algorithm.

CHAPTER IV

EXTENSIONS OF THE TRAVELING SALESMAN PROBLEM

The extensions of the traveling salesman problem which will be discussed in this chapter arise most obviously in problems dealing with the sequencing of jobs through a single processing facility. Suppose that one wishes to determine the minimum cost sequence of n jobs through a single machine. The costs associated with the processing of each job are assumed to be fixed and therefore are not affected by the particular sequence in which the jobs are processed. However, the cost of setting up the machine for any particular job is dependent upon which job immediately preceded that job on the machine. This sequencing problem is an example of the traveling salesman problem which was formulated in Equations (3.7) through (3.11) of Chapter III.

The first extension to be considered will occur when the job sequence is required to maintain some specified pattern; therefore, this problem shall be discussed under the heading of "Pattern Constraints." The other extension is discussed in the section labeled "Due Dates" because it occurs when there are due dates and corresponding late charges assigned to the jobs to be processed. Branch and bound algorithms will be formulated for the solution of both of these sequencing problems.

Pattern Constraints

As discussed above, the term "pattern constraints" refers to the requirement that the job sequence conform to some specified pattern. This term was first used by Eastman (10) to describe the constraint which prohibits subtours in the general traveling salesman problem as well as to describe precedence constraints. Eastman extended his branch and bound algorithm to handle precedence constraints by working first toward a tour solution and then toward a tour solution which did not violate the precedence constraints. In this chapter we shall consider both the subtour constraint and the precedence constraints as one complete set of pattern constraints.

The algorithm of Little, *et al.* (21) has proven to be more efficient for symmetric problems than Eastman's algorithm (see Reference 21); therefore, an algorithm with the same general structure as Little's algorithm will also be formulated for the treatment of this problem.

Consider a problem of the following form:

$$P^{o} \qquad \text{minimize } C^{o}(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.1}$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} = 1, \; j = 1, \ldots, n \tag{4.2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1, \ldots, n \tag{4.3}$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j \tag{4.4}$$

$$\text{no subtours} \tag{4.5}$$

This problem is the standard traveling salesman problem discussed in Chapter III. Suppose that we add to this problem the constraint that job *a* precede job *b*, which must precede job *c* in the solution sequence, i.e.,

$$a > b > c \tag{4.6}$$

The problem described by Equations (4.1) through (4.6) is a "constrained traveling salesman problem." However, since the no sub-tour constraint may also be considered as a pattern constraint, it will be convenient to think of constraints (4.5) and (4.6) as comprising a set of pattern constraints for this problem. This problem may be solved by either of the following branch and bound algorithms.

Extension of Eastman's Algorithm

To solve the problem P^o defined by Equations (4.1) through (4.6), one may use a branch and bound algorithm which has a general stage problem identical to that of Eastman's algorithm described in Chapter III. The problem P^k defined by condition 0-1 is of the following form.

$$P^k \qquad \text{minimize } C^k(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} = 1, \; j = 1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1,\ldots,n$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j$$

$$x_{ij} = 0 \text{ for } x_{ij} \in R^k$$

As with Eastman's algorithm, we set $R^1 = \phi$; however, for this particular precedence constraint, we may set $R^1 = \{x_{ac}, x_{ba}, x_{cb}\}$. This is not necessary, but it will give us a better starting point because these variables are obviously required to be zero.

The branching rule for this algorithm is identical to that given for Eastman's algorithm in Chapter III except that now a violation of either of the pattern constraints (4.5) and (4.6) will be used to initiate branching. Suppose that X^{k*} violates either constraint (4.5) or constraint (4.6), or possibly both constraints. Let S_b be the shortest subtour and S_v be the shortest partial sequence which violates the precedence constraint; for example, $S_v = \{(a,i),(i,j),(j,c)\}$. Choose the shorter of these two and branch on that violation using Eastman's branching rule. This algorithm is illustrated by an example problem in Appendix A.

Extension of Little's Algorithm

Again it is desired to solve the problem P^o described by Equations (4.1) through (4.6). Now we shall approach the problem by extending Little's algorithm in the same fashion that Eastman's algorithm was extended.

For the general stage problem, let P^k be as defined by condition 0-1. It is of the form

P^k minimize $C^k(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

S.T. $\sum_{i=1}^{n} x_{ij} \leq 1, \; j = 1, \ldots, n$

$\sum_{j=1}^{n} x_{ij} \leq 1, \; i = 1, \ldots, n$

$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = m^k$

$x_{ij} = 0$ or 1, all i and j

no subtours

$a > b > c$

$x_{ij} = 0$ for $x_{ij} \in R^k$

$x_{ij} = 1$ for $x_{ij} \in Q^k$

where for problem P^1, $m^1 = 1$ and $Q^1 = \phi$. As with the extension of Eastman's algorithm we can set $R^1 = \{x_{ac}, x_{ba}, x_{cb}\}$. The branching rule for this algorithm is identical to that given in Chapter III for Little's algorithm. It should be noted that the precedence constraint is now active for every problem P^j. For example, suppose problem P^j has $x_{aj} \in Q^j$, i.e., $x_{aj} = 1$; then $x_{jc} = 1$ would be an infeasible solution to P^j. This algorithm is illustrated by an example problem in Appendix B.

This section has examined an extension of the traveling salesman problem which is actually a traveling salesman problem with additional constraints. In the next section we shall examine a sequencing problem which has the basic structure of the traveling salesman problem, with the objective function changed to include costs other than sequence-dependent setup costs. A branch and bound algorithm similar to Little's algorithm will be developed for the solution of this problem.

Due Dates

Consider the problem of finding the minimum cost processing sequence for a set of jobs $D = \{1,\ldots,n\}$ which are to be processed by a single facility. Let these jobs arrive simultaneously at time $t = o$. For each job i, a "due date" a_i is specified. The jobs are partitioned into two mutually exclusive and collectively exhaustive sets A and B. The processing sequence will determine an actual completion time F_i for each job i, and a penalty charge will be incurred by job i if F_i is greater than a_i. This penalty charge will be L_i for $i \varepsilon A$, or $L_i (F_i - a_i)$ for $i \varepsilon B$.

Let the setup times and setup costs for this problem be sequence dependent, i.e., a setup time of t_{ij} and a setup cost of c_{ij} are incurred when job i immediately precedes job j in the processing sequence The setup times and costs, as well as the processing times PT_i and costs PC_i, $i \varepsilon D$, are deterministic and known.

The processing sequence for this problem shall be required to be a traveling salesman's tour. If job I is specified as the first job to be processed, then the facility must be returned to the state necessary for processing job I after the n jobs have been processed.

Model for Due Dates Problem

An order of processing for any set of jobs, starting with job I and ending with job p, can be denoted by a sequence $S_p = (I,i,j,\ldots,p)$ which may or may not include all jobs $i \in D$. This sequence can be defined by the variables x_{ij} as follows:

$$x_{ij} = \begin{cases} 1 & \text{if job } i \text{ immediately precedes job } j \text{ in } S_p \\ 0 & \text{otherwise} \end{cases}$$

Let the "cost" of a sequence $C(S_p)$ be defined by

$$C(S_p) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{n} PC_i + \sum_{i=1}^{n} L_i y_i \tag{4.7}$$

where

$$y_i = \begin{cases} 0 & \text{if } a_i \geq \hat{F}_i \\ 1 & \text{if } a_i < \hat{F}_i \end{cases} \tag{4.8}$$

for $i \in A$, or

$$y_i = \begin{cases} 0 & \text{if } a_i \geq \hat{F}_i \\ (\hat{F}_i - a_i) & \text{if } a_i < \hat{F}_i \end{cases} \tag{4.9}$$

for $i \in B$. The number $\hat{F}_i$ associated with each job shall be defined by

$$\hat{F}_i = \sum_{k \varepsilon S_i} PT_k + \sum_{k,j \varepsilon S_i} t_{kj} x_{kj} \tag{4.10}$$

for $i \varepsilon S_p$, or

$$\hat{F}_i = \sum_{k \varepsilon S_p} PT_k + \sum_{k,j \varepsilon S_p} t_{kj} x_{kj} + PT_i + t_{pi} \tag{4.11}$$

for $i \notin S_p$, where $S_i = (I,\ldots,i)$, i.e., S_i is a subsequence of S_p.

Equation (4.10) gives the actual time when processing will be completed on job i for $i \varepsilon S_p$, i.e., $\hat{F}_i = F_i$. Equation (4.11) gives a lower bound for the completion time on job i for $i \notin S_p$, $i \varepsilon D$.

Any feasible solution sequence for the due dates problem will contain all of the n jobs, and Equation (4.7) will give the actual cost of the feasible processing sequence.

Problem P^o can now be stated in the following form.

P^o minimize $C(S_p)$ (4.12)

$$\text{S.T.} \quad \sum_{i=1}^{n} x_{ij} = 1, \; j = 1,\ldots,n \tag{4.13}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1,\ldots,n \tag{4.14}$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j \tag{4.15}$$

$$\text{no subtours} \tag{4.16}$$

It may be noted that constraints (4.13) and (4.14) imply that S_p should contain all jobs $i \in D$.

Solution Algorithm for Due Dates Problem

The problem P^o, described by Equations (4.12) through (4.16), can be solved by the following branch and bound algorithm. Let problem P^k, $k \in P$ as defined by condition 0-1 of Theorem 1, be as follows:

$$P^k \qquad \text{minimize } C(S_p)$$

$$\text{S.T.} \qquad \sum_{i=1}^{n} x_{ij} \le 1, \; j = 1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} \le 1, \; i = 1,\ldots,n$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \ge m^k$$

$$x_{ij} = 0 \text{ or } 1, \text{ all } i \text{ and } j$$

no subtours

$$x_{ij} = 0 \text{ for } x_{ij} \in R^k$$

$$x_{ij} = 1 \text{ for } x_{ij} \in Q^k$$

where R^k is the set of variables specified to be equal to zero; Q^k is the set of variables specified to be equal to one, and the cardinality of Q^k is m^k-1.

This general stage problem is almost identical to that given for Little's algorithm in Chapter III. Notice that we are now required to minimize the cost of a sequence which, by definition, is an ordered set of jobs beginning with some designated initial job I.

To initialize this algorithm, the first current set of bounding problems P consists of one problem P^1 with $R^1 = Q^1 = \phi$; hence, $m^1 = 1$ as before. The solution to P^1 is $X_{Iq} = 1$ for some $q \neq I$, and $X_{ij} = 0$ for all of the remaining variables.

As in Little's algorithm, problem P^1 clearly satisfies the bounding properties B-1 and B-2, and the convergence condition C-1. Also, the general stage problem P^k, as defined by condition 0-1, satisfies B-1, B-2, and C-3 The branching rule for this algorithm is identical to the one used for Little's algorithm; therefore, the convergence conditions C-2 and C-3 are satisfied, and the algorithm will converge to an optimal solution to P^o in a finite number of steps. This algorithm is illustrated by an example problem in Appendix C.

CHAPTER V

DISCUSSION AND RECOMMENDATION

The general branch and bound approach of Lawler and Wood (19) has been extended in this report to include some sufficient conditions for convergence. These conditions are more general than those of Balas (1) because they make no special assumptions about the form of the original problem P^o, whereas, Balas required that P^o have a finite solution space. Also, the optimality condition 0-3 relaxes Lawler and Wood's equality to a weak inequality.

Lawler and Wood's formulation of the branch and bound approach is in terms of bounding problems. Chapter III presents the form of these bounding problems for Eastman's and Little's algorithms. It is also shown that each of these algorithms satisfies the hypotheses of Theorem 2, which insure convergence

This formulation of the branch and bound approach has readily permitted the application of these algorithms, in Chapter IV, to the traveling salesman problem with pattern constraints in addition to the "no subtour" pattern constraint. Also in Chapter IV, a sequencing problem with change-over and "late penalty costs" was formulated. A branch and bound algorithm with the same basic structure as Little's algorithm was developed for the solution of this problem.

The original objective of this study was to develop solution methods for the problems of Chapter IV. In pursuing this objective, the generality and flexibility of the branch and bound approach became apparent. An attempt was made to determine the necessary and sufficient conditions for the finite convergence of any branch and bound algorithm. This attempt was unsuccessful, although some sufficient conditions for finite convergence were developed.

It is the author's opinion that further study should be directed toward developing these necessary and sufficient conditions for convergence. They would greatly enhance the general applicability of the branch and bound approach to problem solving.

APPENDICES

APPENDIX A

APPLICATION OF EASTMAN'S ALGORITHM

In order to illustrate the application of Eastman's branch and bound algorithm (10), we shall consider a traveling salesman problem with the following cost matrix

	1	2	3	4	5
1	∞	8	9	8	12
2	7	∞	4	11	10
3	5	4	∞	11	8
4	6	9	6	∞	7
5	9	7	8	3	∞

and with the additional pattern constraint 2 > 5 > 3. The solution of this problem is outlined below.

1. P^1 is an assignment problem with $R^1 = \{x_{23}, x_{52}, x_{35}\}$. The solution to P^1 is $x_{13} = x_{32} = x_{21} = x_{45} = x_{54} = 1$, all other $x_{ij} = 0$. This solution contains two subtours. The shortest of these is $S_b = \{(4,5),(5,4)\}$; therefore, we branch on P^1 and create two new problems P^2 and P^3.

2. The current set P now is $P = \{P^2, P^3\}$. $R^2 = \{x_{23}, x_{52}, x_{35}, x_{45}\}$, and the solution to P^2 is $x_{15} = x_{54} = x_{43} = x_{32} = x_{21} = 1$, all

other $x_{ij} = 0$; therefore, $C^2(X^{2*}) = 32$. The solution to P^3 is $x_{14} = x_{45} = x_{53} = x_{32} = x_{21} = 1$, all other $x_{ij} = 0$; therefore, $C^3(X^{3*}) = 34$. Problem $P^2 = P^k$, where P^k is defined by condition 0-1 of Theorem 1, because $C^2(X^{2*}) = 32 \leq C^3(X^{3*})$. The solution X^{2*} is a tour which does not violate the precedence constraint, and $C^2(X^{2*}) = C^o(X^{2*})$; hence, conditions 0-2 and 0-3 of Theorem 1 are satisfied and $X^{o*} = X^{2*}$.

APPENDIX B

APPLICATION OF LITTLE'S, *et al.*, ALGORITHM

Now we shall use Little's (21) algorithm to solve the problem of Appendix A. The solution procedure is illustrated below.

1. The reduced cost matrix for P^1 is

	1	2	3	4	5
1	∞	$^{0}\,0$	1	$^{0}\,0$	3
2	$^{2}\,0$	∞		4	2
3	1	$^{1}\,0$	∞	7	
4	$^{0}\,0$	3	$^{1}\,0$	∞	$^{2}\,0$
5	6		5	$^{5}\,0$	∞

where indicates that x_{ij} is excluded because $x_{ij} = 1$ would cause a precedence violation. The small number in the top left corner of each box is the amount by which the matrix could be reduced if c_{ij} for that box is removed from the problem. Little's heuristic procedure for breaking ties in the solution of any problem is to pick that $x_{pq} = 1$ which will cause the greatest reduction of the matrix for the new bounding problem which has $x_{pq} \in R^j$, i.e., $x_{pq} = 0$. Therefore, the solution to P^1 is $x_{54} = 1$, all other $x_{ij} = 0$; $C^1(X^{1*}) = c_{pq}$ + sum of the row and column reductions = 29.

2. The cost matrices for problems P^2 and P^3 are given below. Let $\boxed{\diagdown}$ indicate that x_{ij} is excluded because it would cause a subtour, and $\boxed{-}$ indicates an exclusion due to the other constraints of the problem. Let $\boxed{X}$ imply $x_{ij} \in R^j$ and $\boxed{\square}$ imply $x_{ij} \in Q^j$.

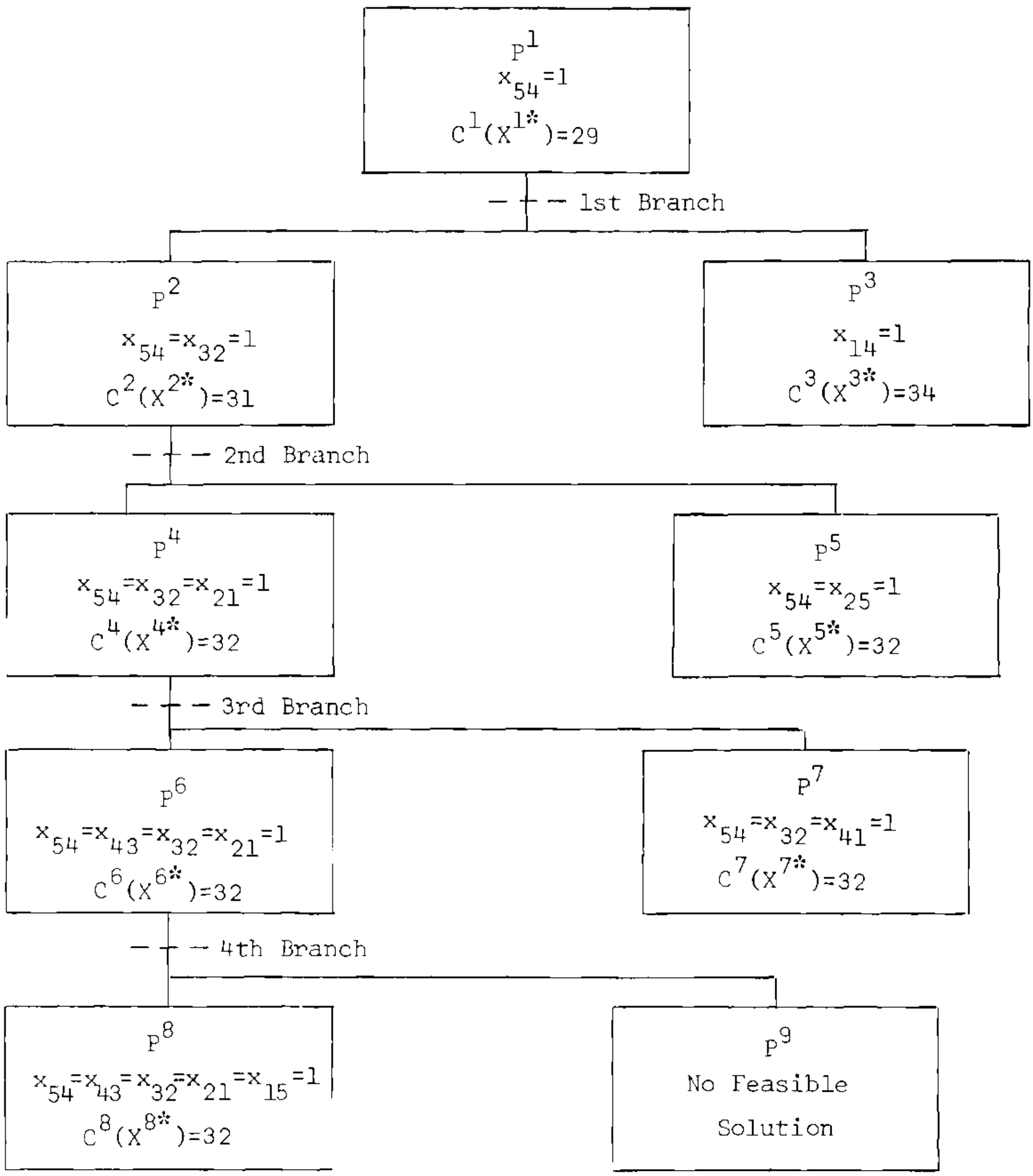
P^2

	1	2	3	4	5
1	∞	${}^{1}\,0$	1	-	1
2	${}^{0}\,0$	∞		-	${}^{1}\,0$
3	1	${}^{1}\,0$	∞	-	
4	${}^{0}\,0$		${}^{1}\,0$	∞	
5	-		-	$\square$	∞

P^3

	1	2	3	4	5
1	∞	${}^{0}\,0$	1	${}^{4}\,0$	3
2	${}^{2}\,0$	∞		4	2
3	1	${}^{1}\,0$	∞	7	
4	${}^{0}\,0$	3	${}^{0}\,0$	∞	${}^{2}\,0$
5	1		${}^{1}\,0$	X	∞

The current set is $P = \{P^2, P^3\}$ at this stage. The solution to P^2 is $x_{54} = x_{32} = 1$, all other $x_{ij} = 0$, and $C^2(X^2) = 31$. The solution to P^3 is $x_{14} = 1$, all other $x_{ij} = 0$, and $C^3(X^3) = 34$. Therefore, we branch from problem P^2. The entire solution process creates the tree shown in Figure 1 below.



Final Current Set $P = \{P^3, P^5, P^7, P^8, P^9\}$;
$C^8(X^{8*}) = \min C^j(X^{j*})$; $C^8(X^{8*}) = C^o(X^{8*})$; $X^{8*} \varepsilon D^o$;
$\therefore X^{o*} = X^{8*}$.

Figure 1. Solution Tree for Little's Algorithm

APPENDIX C

APPLICATION OF DUE DATES ALGORITHM

Consider a sequencing problem as follows:

JOB	1	2	3	4
PT_i	6	4	2	3
a_i	14	22	18	7
L_i	3	2	12	8

$$\sum_{i=1}^{4} PC_i = 27$$

$$A = \{2,3,4\}$$

$$B = \{1\}$$

	1	2	3	4
1	∞	2 3	3 2	1 2
2	5 4	∞	3 1	3 6
3	2 5	6 3	∞	4 5
4	1 2	5 4	2 1	∞

where c_{ij} and t_{ij} are elements of the above matrix such that $\boxed{\frac{c_{ij}}{t_{ij}}}$.
Let the initial job I be job 4. For problem one we pick some $x_{4j} = 1$ such that $C(S_j)$ is minimized. The calculations for P^1 are given in Figure 2 below.

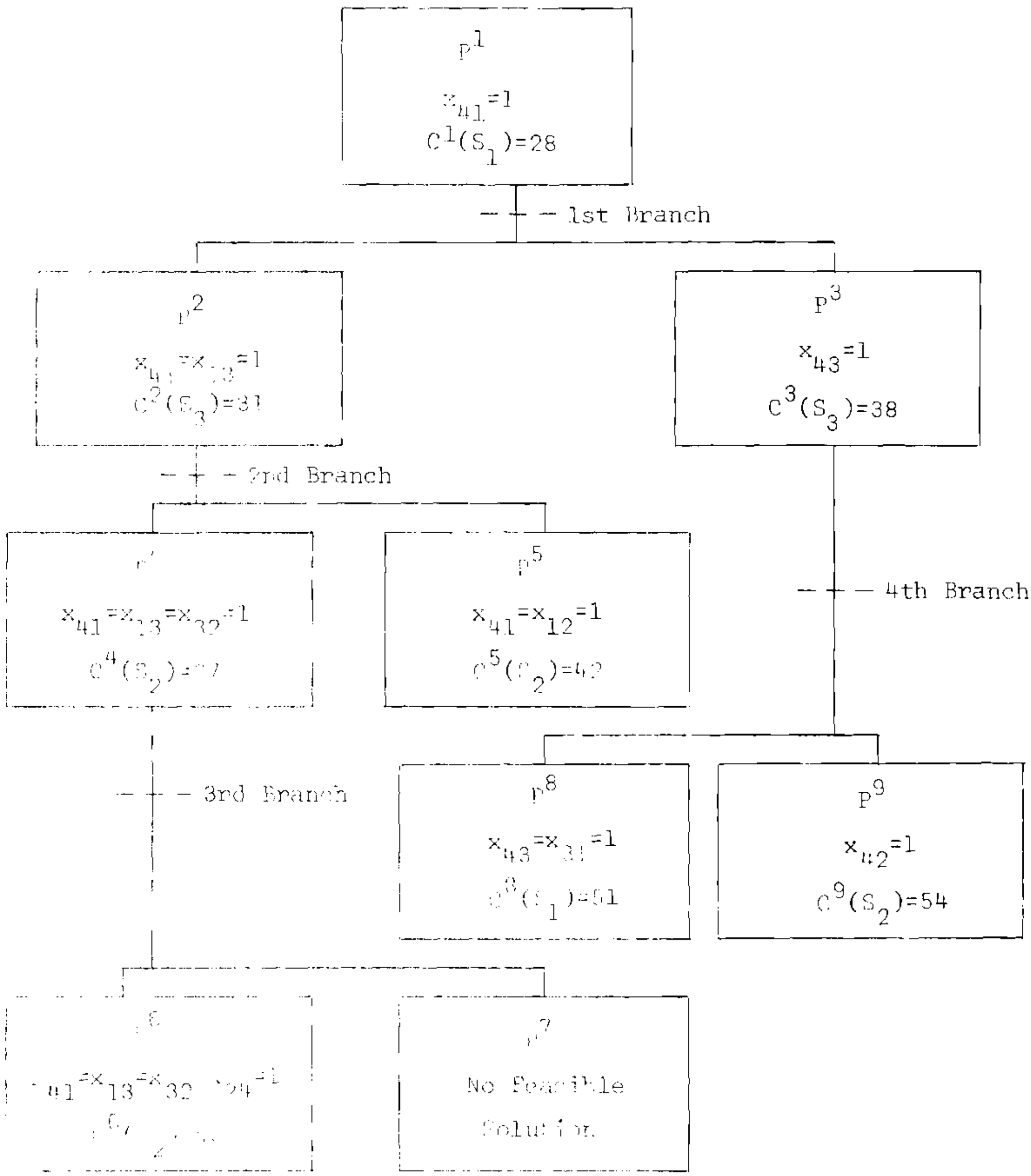
The solution to P^1 is to have $x_{41} = 1$, all other $x_{ij} = 0$; so we branch on x_{41} to create new problems P^2 and P^3 in the same way that we did for Little's algorithm. Notice that for problem P^3, which has $x_{41} \varepsilon R^3$, i.e., $x_{41} = 0$, we do not have to re-calculate the solution. The calculations done in P^1 are still good for $x_{42} = 1$ or $x_{43} = 1$.

This algorithm continues constructing problems and branching in this manner until X^{o*} is found. The solution tree for this problem is shown in Figure 3.

If $x_{41} = 1$	If $x_{42} = 1$	If $x_{43} = 1$
$\hat{F}_1 = 3 + 2 + 6 = 11$	$\hat{F}_2 = 3 + 4 + 4 = 11$	$\hat{F}_3 = 3 + 1 + 2 = 6$
$\hat{F}_2 = 11 + 4 + 3 = 18$	$\hat{F}_1 = 11 + 6 + 4 = 21$	$\hat{F}_1 = 6 + 6 + 5 = 17$
$\hat{F}_3 = 11 + 2 + 2 = 15$	$\hat{F}_3 = 11 + 2 + 1 = 14$	$\hat{F}_2 = 6 + 4 + 3 = 13$
$C(S_4) = 27$	$C(S_4) = 27$	$C(S_4) = 27$
$C_{41} = 1$	$C_{42} = 5$	$C_{43} = 2$
$L_1y_1 = 0$	$L_1y_1 = 21$	$L_1y_1 = 9$
$L_2y_2 = 0$	$L_2y_2 = 0$	$L_2y_2 = 0$
$L_3y_3 = 0$	$L_3y_3 = 0$	$L_3y_3 = 0$
$C(S_1) = 28$	$C(S_2) = 53$	$C(S_3) = 38$

The solution to problem P^1 is $x_{41} = 1$ and all other $x_{ij} = 0$. $C(S_p) = C(S_1) = 28$.

Figure 2. Calculations for Problem P^1



Final Current Set $P = \{P^6, P^7, P^8, P^9\}$

$C^6(S_2) = \min_{j \in P} C^j(S_p)$; $C^6(X^{6*}) = C^0(X^{6*})$; $X^{6*} \varepsilon D^0$

$\therefore X^{6*} = X^{0*}$

Figure 3. Solution Tree for Due Dates Algorithm

BIBLIOGRAPHY

1. Balas, E., "A Note on the Branch-and-Bound Principle," *Operations Research 16,* 442-445, 1968.

2. Barachet, L. L., "Graphic Solution of the Traveling Salesman Problem," *Operations Research 5,* 841-845, 1957.

3. Bellman, R., "Dynamic Programming Treatment of the Traveling Salesman Problem," *Journal of the Association for Computing Machinery 9,* 61-63, 1962.

4. Bellmore, M., and G. L. Nemhauser, "The Traveling Salesman Problem: A Survey," *Operations Research 16,* 538-558, 1968.

5. Conway, R. W., W. L. Maxwell, and L. W. Miller, *Theory of Scheduling,* Addison-Wesley Co., 1967.

6. Croes, G. A., "A Method for Solving Traveling Salesman Problems," *Operations Research 6,* 791-812, 1958.

7. Dacey, M. F., "Selection of an Initial Solution for the Traveling Salesman Problem," *Operations Research 8,* 133-134, 1960.

8. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem," *Operations Research 7,* 58-66, 1959.

9. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of a Large Scale Traveling Salesman Problem," *Operations Research 2,* 373-410, 1954.

10. Eastman, W. L., "Linear Programming with Pattern Constraints," Ph.D. Dissertation, Harvard, 1958.

11. Elmaghraby, S. E., "The Machine Sequencing Problem—Review and Extensions," *Naval Research Logistics Quarterly 15,* 205-232, 1968.

12. Flood, M. M., "The Traveling Salesman Problem," *Operations Research 4,* 61-75, 1956.

13. Gavett, W. J., "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility," *Management Science 11,* 476-484, 1965.

14. Gilmore, P. C., and R. E. Gomory, "Sequencing a One-State Variable Machine: A Solvable Case of the Traveling Salesman Problem," *Operations Research 12,* 655-679, 1964.

15. Gonzalez, R. H., "Solution of the Traveling Salesman Problem by Dynamic Programming on the Hypercube," Interim Technical Report No. 18, OR Center, M.I.T., 1962.

16. Hatfield, D. J., and J. F. Pierce, "Production Sequencing by Combinatorial Programming," IBM Cambridge Scientific Center, Cambridge, Mass., 1966.

17. Held, M., and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," *SIAM Journal 10,* 196-210, 1962.

18. Karg, L. L., and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems," *Management Science 10,* 225-248, 1964.

19. Lawler, E. L., and D. E. Wood, "Branch and Bound Methods: A Survey," *Operations Research 14,* 699-719, 1966.

20. Lin, S., "Computer Solution of the Traveling Salesman Problem," *Bell System Technical Journal 44,* 2245-2269, 1965.

21. Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem," *Operations Research 11,* 979-989, 1963.

22. Miller, C. E., A. W. Tucker, and R. A. Zemlin, "Integer Programming Formulations and Traveling Salesman Problems," *Journal of the Association for Computing Machinery 7,* 326-329, 1960.

23. Reiter, S. and G. Sherman, "Discrete Optimizing," *SIAM Journal 13,* 864-889, 1965.

24. Roberts, S. M., and B. Flores, "An Engineering Approach to the Traveling Salesman Problem," *Management Science 13,* 269-288, 1966.

25. Rothkopf, M., "The Traveling Salesman Problem: On the Reduction of Certain Large Problems to Smaller Ones," *Operations Research 14,* 532-533, 1966.

26. Shapiro, D., "Algorithms for the Solution of the Optimal Cost and Bottleneck Traveling Salesman Problems," Sc. D. Thesis, Washington University, St. Louis, 1966.

27. Spinner, A. H., "Sequencing Theory Development to Date," *Naval Research Logistics Quarterly 15,* 319-330, 1968.